

REPORT DOCUMENTATION PAGE				Form Approved OMB NO. 0704-0188	
<p>The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA, 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.</p> <p>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</p>					
1. REPORT DATE (DD-MM-YYYY)		2. REPORT TYPE		3. DATES COVERED (From - To)	
		Technical Report		-	
4. TITLE AND SUBTITLE Improved graph clustering			5a. CONTRACT NUMBER		
			W911NF-11-1-0265		
			5b. GRANT NUMBER		
			5c. PROGRAM ELEMENT NUMBER		
			611102		
6. AUTHORS Yudong Chen, Sujay Sanghavi			5d. PROJECT NUMBER		
			5e. TASK NUMBER		
			5f. WORK UNIT NUMBER		
7. PERFORMING ORGANIZATION NAMES AND ADDRESSES				8. PERFORMING ORGANIZATION REPORT NUMBER	
University of Texas at Austin 101 East 27th Street Suite 5.300 Austin, TX 78712 -1539					
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)  U.S. Army Research Office P.O. Box 12211 Research Triangle Park, NC 27709-2211				10. SPONSOR/MONITOR'S ACRONYM(S) ARO	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S) 59441-NS.5	
12. DISTRIBUTION AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.					
13. SUPPLEMENTARY NOTES The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.					
14. ABSTRACT Graph clustering involves the task of partitioning nodes, so that the edge density is higher within partitions as opposed to across partitions. A natural, classic and popular statistical setting for evaluating solutions to this problem is the stochastic block model, also referred to as the planted partition model. In this paper we present a new algorithm - a convexified version of Maximum Likelihood - for graph clustering. We show that, in the classic stochastic block model setting, it outperforms all existing methods by polynomial					
15. SUBJECT TERMS graph clustering, learning					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Sanjay Shakkottai
UU	UU	UU	UU		19b. TELEPHONE NUMBER
					512-471-5376

## Report Title

Improved graph clustering

### ABSTRACT

Graph clustering involves the task of partitioning nodes, so that the edge density is higher within partitions as opposed to across partitions. A natural, classic and popular statistical setting for evaluating solutions to this problem is the stochastic block model, also referred to as the planted partition model.

In this paper we present a new algorithm - a convexified version of Maximum Likelihood - for graph clustering. We show that, in the classic stochastic block model setting, it outperforms all existing methods by polynomial factors; in fact, it is within logarithmic factors of known lower bounds for spectral methods.

We then show that this guarantee carries over to a more general semi-random extension of the stochastic block model; our method can handle settings of heterogeneous degree distributions, unequal cluster sizes, outlier nodes, planted k-cliques etc.

# Improved Graph Clustering

Yudong Chen, Sujay Sanghavi, *Member, IEEE*, and Huan Xu

## Abstract

Graph clustering involves the task of partitioning nodes, so that the edge density is higher within partitions as opposed to across partitions. A natural, classic and popular statistical setting for evaluating solutions to this problem is the stochastic block model, also referred to as the planted partition model.

In this paper we present a new algorithm - a convexified version of Maximum Likelihood - for graph clustering. We show that, in the classic stochastic block model setting, it outperforms all existing methods by polynomial factors; in fact, it is within logarithmic factors of known lower bounds for spectral methods.

We then show that this guarantee carries over to a more general semi-random extension of the stochastic block model; our method can handle settings of heterogeneous degree distributions, unequal cluster sizes, outlier nodes, planted  $k$ -cliques etc.

## I. INTRODUCTION

This paper proposes a new algorithm for the following task: given an undirected unweighted graph, assign the nodes into disjoint clusters so that the density of edges within clusters is higher than the edges across clusters. Clustering arises in applications such as a community detection, user profiling, link prediction, collaborative filtering etc. In these applications, one is often given as input a set of similarity relationships (either “1” or “0”) and the goal is to identify groups of similar objects. For example, given the friendship relations on Facebook, one would like to detect tightly connected communities, which is useful for subsequent tasks like customized recommendation and advertisement.

Graphs in modern applications have several characteristics that complicate graph clustering:

- *Small density gap*: the edge density across clusters is only a small additive or multiplicative factor different from within clusters;
- *Sparsity*: the graph is overall very sparse even within clusters;
- *Outliers*: there may exist nodes that do not belong to any clusters and are loosely connected to the rest of the graph;
- *High dimensionality*: the number of clusters may grow unbounded as a function of the number of nodes  $n$ , which means the sizes of the clusters can be vanishingly small compared to  $n$ ;
- *Heterogeneity*: the cluster sizes, node degrees and edge densities may be non-uniform; there may even exist edges that are not well-modeled by probabilistic distributions as well as hierarchical cluster structures.

Various large modern datasets and graphs have such characteristics [1, 2]; examples include the web graph, social graphs of various social networks etc. As has been well-recognized, these characteristics make clustering more difficult. When the difference between in-cluster and across-cluster edge densities is small, the clustering structure is less significant and thus harder to detect. Sparsity further reduces the amount of information and makes the problem noisier. In the high dimensional regime, there are many small clusters, which are easy to lose in the noise. Heterogeneous and non-random structures in the graphs foil many algorithms that otherwise perform well. Finally, the existence of hierarchy and outliers renders

The work of Y. Chen was supported by NSF grant EECS-1056028 and DTRA grant HDTRA 1-08-0029. The work of S. Sanghavi was supported by NSF grant 1017525, an Army Research Office (ARO) grant W911NF-11-1-0265, and a DTRA Young Investigator award. The work of H. Xu was partially supported by the Ministry of Education of Singapore through AcRF Tier Two grant R-265-000-443-112 and NUS startup grant R-265-000-384-133.

Y. Chen and S. Sanghavi are with the Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, TX 78712 USA. H. Xu is with the Department of Mechanical Engineering, National University of Singapore, 9 Engineering Drive 1, Singapore 117575, Singapore. (e-mail: ydchen@utexas.edu, sanghavi@mail.utexas.edu, mpexuh@nus.edu.sg)

many existing algorithms and theoretical results inapplicable, as they fix the number of clusters a priori and force each node to be assigned to a cluster. It is desirable to design an algorithm that can handle all these issues in a principled manner.

### A. Our Contributions

Our *algorithmic contribution* is a new method for unweighted graph clustering. It is motivated by the maximum-likelihood estimator for the classical Stochastic Blockmodel [3] (a.k.a. the Planted Partition Model [4]) for random clustered graphs. In particular, we show that this maximum-likelihood estimator can be written as a linear objective over combinatorial constraints; our algorithm is a convex relaxation of these constraints, yielding a convex program overall. While this is the motivation, it performs well – both in theory and practice – in settings that are not just the standard stochastic blockmodel.

Our *main analytical result* in this paper is theoretical guarantees on its performance; we study it in a *semi-random generalized stochastic blockmodel*. This model generalizes not only the standard stochastic blockmodel and planted partition model, but many other classical planted models including the planted  $k$ -clique model [5, 6], the planted coloring model [7, 8] and their semi-random variants [9, 10, 11]. Our main result gives the condition (as a function of the in/cross-cluster edge densities  $p$  and  $q$ , density gap  $|p - q|$ , cluster sizes  $K$  and numbers of inliers/outliers  $n_1$  and  $n_2$ ) under which our algorithm is guaranteed to recover the ground-truth clustering. When  $p > q$ , the condition reads

$$p - q = \Omega \left( \frac{\sqrt{p(1-q)}(n_1 + n_2)}{K} \right);$$

here all the parameters are allowed to scale with  $n = n_1 + n_2$ , the total number of nodes. An analogue result holds for  $p < q$ .

While the planted and stochastic block models have a rich literature, this single result shows that our algorithm outperforms every existing method for the standard planted partition/ $k$ -clique/noisy-coloring models, and matches them (up to at most logarithmic factors) in all other cases, in the sense that our algorithm succeeds for a bigger range of the parameters. In fact, there is evidence indicating that we are close to the boundary at which any polynomial-time algorithm can be expected to work. Moreover, the proof for our main theorem is relatively simple, relying only on standard concentration results. Our *simulation study* supports our theoretic finding, that the proposed method is effective in clustering noisy graphs and outperforms existing methods.

The rest of the paper is organized as follows: Section I-B provides an overview of related work; Section II presents our algorithm; Section III describes the Semi-Random Generalized Stochastic Blockmodel (which is a generalization of the standard stochastic blockmodel, one that allows the modeling of the issues mentioned above); Section IV presents the main results – a performance analysis of our algorithm for the generalized stochastic blockmodel and a detailed comparison to the existing literature on this problem; Section V provides simulation results; finally, the proof of our theoretic results is given in Sections VI to IX.

### B. Related Work

The general field of clustering, or even graph clustering, is too vast for a detailed survey here; we focus on the most related threads, and therein too primarily on work which provides analytical guarantees on the resulting algorithms.

**Stochastic block models:** Also called “planted models” [3, 4], these are perhaps the most natural random clustered graph models. In the simplest or standard setting, nodes are partitioned into disjoint equal-sized subsets (called the underlying clusters), and then edges are generated independently and at random, with the probability  $p$  of an edge between two nodes in the same cluster higher than the probability  $q$  when the two nodes are in different clusters. The algorithmic clustering task in this setting is to recover

TABLE I  
COMPARISON WITH LITERATURE FOR THE STANDARD STOCHASTIC BLOCKMODEL

Paper	Cluster size $K$	Density gap $p - q$	Sparsity $p$
Boppana (1987) [15]	$n/2$	$\tilde{\Omega}(\sqrt{\frac{p}{n}})^a$	$\tilde{\Omega}(\frac{1}{n})$
Jerrum et al. (1998) [16]	$n/2$	$\tilde{\Omega}\left(\frac{1}{n^{1/6-\epsilon}}\right)$	$\tilde{\Omega}\left(n^{1/6-\epsilon}\right)$
Condon et al. (2001) [3]	$\Theta(n)$	$\tilde{\Omega}\left(\frac{1}{n^{1/2-\epsilon}}\right)$	$\tilde{\Omega}\left(n^{1/2-\epsilon}\right)$
Carson et al. (2001) [17]	$n/2$	$\tilde{\Omega}\left(\sqrt{\frac{p}{n}}\right)$	$\tilde{\Omega}\left(\frac{1}{n}\right)$
Feige et al. (2001) [10]	$n/2$	$\tilde{\Omega}\left(\sqrt{\frac{p}{n}}\right)$	$\tilde{\Omega}\left(\frac{1}{n}\right)$
McSherry (2001) [5]	$\Omega(n^{2/3})$	$\tilde{\Omega}\left(\sqrt{\frac{pn^2}{K^3}}\right)$	$\tilde{\Omega}\left(\frac{n^2}{K^3}\right)$
Bollobas (2004) [9]	$\Theta(n)$	$\tilde{\Omega}\left(\sqrt{\frac{q}{n}} \vee \frac{1}{n}\right)$	$\tilde{\Omega}\left(\frac{1}{n}\right)$
Giesen et al. (2005) [18]	$\Omega(\sqrt{n})$	$\tilde{\Omega}\left(\frac{\sqrt{n}}{K}\right)$	$\tilde{\Omega}\left(\frac{\sqrt{n}}{K}\right)$
Shamir (2007) [19]	$\Omega(\sqrt{n} \log n)$	$\tilde{\Omega}\left(\frac{\sqrt{n}}{K}\right)$	$\tilde{\Omega}\left(\frac{\sqrt{n}}{K}\right)$
Coja-Oghlan (2010) [20]	$\Omega(n^{4/5})$	$\tilde{\Omega}\left(\sqrt{\frac{pn^4}{K^5}}\right)$	$\tilde{\Omega}\left(\frac{n^4}{K^5}\right)$
Rohe et al. (2011) [21]	$\Omega((n \log n)^{2/3})$	$\tilde{\Omega}\left(\frac{n^{1/2}}{K^{3/4}}\right)$	$\tilde{\Omega}\left(\frac{1}{\sqrt{\log n}}\right)$
Oymak et al. (2011) [22]	$\Omega(\sqrt{n})$	$\tilde{\Omega}\left(\frac{\sqrt{n}}{K}\right)$	$\tilde{\Omega}\left(\frac{\sqrt{n}}{K}\right)$
Chaudhuri et al. (2012) [12]	$\Omega(\sqrt{n \log n})$	$\tilde{\Omega}\left(\frac{\sqrt{n}}{K}\right)$	$\tilde{\Omega}\left(\frac{\sqrt{n}}{K}\right)$
Ames (2012) [23]	$\Omega(\sqrt{n})$	$\tilde{\Omega}\left(\frac{\sqrt{n}}{K}\right)$	$\tilde{\Omega}\left(\frac{\sqrt{n}}{K}\right)$
Our result	$\Omega(\sqrt{n})$	$\tilde{\Omega}\left(\frac{\sqrt{pn}}{K}\right)$	$\tilde{\Omega}\left(\frac{n}{K^2}\right)$

To facilitate direct comparison, this table specializes some of the results to the case where every underlying partition (i.e. cluster) is of the same size  $K$ , and the in/cross-cluster edge probabilities are uniformly  $p$  and  $q$ . Some of the algorithms above need this assumption, and some – like ours – do not.

<sup>a</sup>The soft  $\tilde{\Omega}(\cdot)$  notation ignores log factors.

the underlying clusters given the graph. The parameters  $p, q$  and the size  $K$  of the smallest cluster typically govern whether an algorithm can do this clustering, or not.

There is now a long line of analytical work on stochastic block models; we focus on methods that allow for exact recovery (i.e. every node is correctly classified), and summarize the conditions required by known methods in Table I. As can be seen, we improve over all existing methods by polynomial factors. In addition, and as opposed to several of these methods, we can handle outliers, heterogeneity, hierarchy in clustering etc. A complimentary line of work has investigated **lower bounds** in this setting; i.e., for what values/scalings of  $p, q$  and  $K$  is it *not* possible (either for any algorithm, or for any polynomial time algorithm) to recover the underlying clusters [12, 13, 14]. We discuss these two lines of work in more details in the main results section.

**Convex methods for matrix decomposition:** Our method is related to recent literature on the recovery of low-rank matrices using convex optimization, and in particular the recovery of such matrices from “sparse” perturbations (i.e. where a fraction of the elements of the low-rank matrix are possibly arbitrarily modified, while others are untouched). Sparse and low-rank matrix decomposition using convex optimization was initiated by [24, 25]; follow-up works [26, 27] have the current state-of-the-art guarantees on this problem, and [28] applies it directly to graph clustering.

The method in this paper is Maximum Likelihood, but it can also be viewed as a weighted version of sparse and low-rank matrix decomposition, with *different elements of the sparse part penalized differently, based on the given input graph*. There is currently no literature or analysis of weighted matrix decomposition; in that sense, while our weights have a natural motivation in our setting, our results are likely to have broader implications, for example robust versions of PCA when not all errors are created equal, but have a corresponding prior.

## II. ALGORITHM

We now describe our algorithm; as mentioned, it is a convex relaxation of Maximum Likelihood (ML) as applied to the standard stochastic blockmodel. So, in what follows, we first develop notation and the exact ML estimator, and then its relaxation.

**ML for the standard stochastic blockmodel:** Recall that in the standard stochastic blockmodel nodes are partitioned into disjoint clusters, and edges in the graph are chosen independently; the probability of an edge between a pair of nodes in the same cluster is  $p$ , and for a pair of nodes in different clusters it is  $q$ . Given the graph, the task is to find the underlying clusters that generated it. To write down the ML estimator for this, let us represent any candidate partition by a corresponding *cluster matrix*  $Y \in \mathbb{R}^{n \times n}$  where  $y_{ij} = 1$  if and only if nodes  $i$  and  $j$  are in the same cluster, and 0 otherwise<sup>1</sup>. Any  $Y$  thus needs to have a block-diagonal structure, with each block being all 1's.

A vanilla ML estimator then involves optimizing a likelihood subject to the combinatorial constraint that the search space is the cluster matrices. Let  $A \in \mathbb{R}^{n \times n}$  be the observed adjacency matrix of the graph<sup>2</sup>; then, the log likelihood function of  $A$  given  $Y$  is

$$\log \mathbb{P}(A|Y) = \log \prod_{(i,j): y_{ij}=1} p^{a_{ij}} (1-p)^{1-a_{ij}} \prod_{(i,j): y_{ij}=0} q^{a_{ij}} (1-q)^{1-a_{ij}}$$

We notice that this can be written, via a re-arrangement of terms, as

$$\log \mathbb{P}(A|Y) = \log \left( \frac{p}{q} \right) \sum_{a_{ij}=1} y_{ij} - \log \left( \frac{1-q}{1-p} \right) \sum_{a_{ij}=0} y_{ij} + C; \quad (1)$$

here  $C$  collects the terms that are independent of  $Y$ . ML estimator would be maximizing the above expression subject to  $Y$  being a cluster matrix. While the objective is a linear function of  $Y$ , this optimization problem is combinatorial due to the requirement that  $Y$  be a cluster matrix (i.e., block-diagonal with each block being all-ones), and is intractable in general.

**Our algorithm:** We obtain a convex and tractable algorithm by replacing the constraint “ $Y$  is a cluster matrix” with (i) constraints  $0 \leq y_{ij} \leq 1$  for all elements  $i, j$ , and (ii) a nuclear norm<sup>3</sup> regularizer  $\|Y\|_*$  in the objective. The latter encourages  $Y$  to be *low-rank*, and is based on the well-established insight that a cluster matrix has low rank – in particular, its rank equals to the number of clusters. (We discuss other related relaxations after we present our algorithm.)

Also notice that the likelihood expression (1) is linear in  $Y$  and only the *ratio* of the two coefficients  $\log(p/q)$  and  $\log((1-q)/(1-p))$  is important. We thus introduce a parameter  $t$  which allows us to choose any ratio. This has the advantage that instead of knowing both  $p$  and  $q$ , we only need to choose one number  $t$  (which should be between  $p$  and  $q$ ; we remark on how to choose  $t$  later). This leads to the following convex formulation:

$$\max_{Y \in \mathbb{R}^{n \times n}} c_A \sum_{a_{ij}=1} y_{ij} - c_{A^c} \sum_{a_{ij}=0} y_{ij} - 48\sqrt{n} \|Y\|_* \quad (2)$$

$$\text{s.t. } 0 \leq y_{ij} \leq 1, \forall i, j. \quad (3)$$

where the weights  $c_A$  and  $c_{A^c}$  are given by

$$c_A = \sqrt{\frac{1-t}{t}} \quad \text{and} \quad c_{A^c} = \sqrt{\frac{t}{1-t}}. \quad (4)$$

Here the factor  $48\sqrt{n}$  balances the contributions of the nuclear norm and the likelihood, and the specific forms of  $c_A$  and  $c_{A^c}$  are derived from our analysis. The optimization problem (2)–(3) is convex and can

<sup>1</sup>We adopt the convention that  $y_{ii} = 1$  for any node  $i$  that belongs to a cluster.

<sup>2</sup>We assume  $a_{ii} = 1$  for all  $i$ .

<sup>3</sup>The nuclear norm of a matrix is the sum of its singular values.

be cast as a Semidefinite Program (SDP) [24, 29]. More importantly, it can be solved using efficient first-order methods for large graphs (see Section V-A).

Our algorithm is given as Algorithm 1. Depending on the given  $A$  and the choice of  $t$ , the optimal solution  $\hat{Y}$  may or may not be a cluster matrix. Checking if a given  $\hat{Y}$  is a cluster matrix can be done easily, e.g., via an SVD, which will also reveal cluster memberships if it is a cluster matrix. If it is not, any one of several rounding/aggregation ideas (e.g., the one in [30]) can be used empirically; we do not delve into this approach in this paper, and simply output failure. In Section IV we provide sufficient conditions under which  $\hat{Y}$  is a cluster matrix, with *no* rounding required.

---

**Algorithm 1** Convex Clustering

---

Input:  $A \in \mathbb{R}^{n \times n}$ ,  $t \in (0, 1)$   
 Solve program (2)–(3) with weights (4). Let  $\hat{Y}$  be an optimal solution.  
**if**  $\hat{Y}$  is a cluster matrix **then**  
     Output cluster memberships obtained from  $\hat{Y}$ .  
**else**  
     Output “Failure”.  
**end if**

---

#### A. Remarks about the Algorithm

Note that while we derive our algorithm from the standard stochastic blockmodel, our analytical results hold in a much more general setting. In practice, one could execute the algorithm (with appropriate choice of  $t$ , and hence  $c_A$  and  $c_{A^c}$ ) on any given graph.

*Tighter relaxations:* The formulation (2)–(3) is not the only way to relax the non-convex ML estimator. Instead of the nuclear norm regularizer, a hard constraint  $\|Y\|_* \leq n$  may be used. One may further replace this constraint with the positive semidefinite constraint  $Y \succeq 0$  and the linear constraints  $y_{ii} = 1$ , both satisfied by any cluster matrix<sup>4</sup>. It is not hard to check that these modifications lead to convex relaxations with smaller feasible sets, so any performance guarantee for our formulation (2)–(3) also applies to these alternative formulations. We choose not to use these potentially tighter relaxations based on the following theoretical and practical considerations: a) These formulations do not work well when the numbers of outliers and clusters are unknown. b) We do not obtain better theoretical guarantees with them. In fact, the work [30] considers these tighter constraints but their exact recovery guarantees are improved by ours. Moreover, as we argue in the next section, our guarantees are likely to be order-wise optimal and thus any alternative convex formulations are unlikely provide significant improvements in a scaling sense. c) Our simpler formulation facilitates efficient solution for large graphs via first-order methods; we describe one such method in Section V-A.

*Choice of  $t$ :* Our algorithm requires an extraneous input  $t$ . For the standard planted  $r$ -clique problem [6, 5] (with  $r$  cliques planted in a random graph  $G_{n,1/2}$ ), one can use  $t = 3/4$  (see Section IV-C2). For the standard stochastic blockmodel (with nodes partitioned into equal-size clusters and edge probabilities being uniformly  $p$  and  $q$  inside and across clusters), the value of  $t$  can be easily computed from data (see Section IV-D). In these cases, our algorithm has no tuning parameters whatsoever and does not require knowledge of the number or sizes of the clusters. For the general setting,  $t$  should be chosen to lie between  $p$  and  $q$ , which now represent the lower/upper bounds for the in/cross-cluster edge densities. As such,  $t$  can be interpreted as the *resolution* of the clustering algorithm. To see this, suppose the clusters have a hierarchical structure, where each big cluster is partitioned into smaller sub-clusters with higher edge densities inside. In this case, it is a priori not clear that which level of clusters, the larger ones or the smaller ones, should be recovered. This ambiguity is resolved by specifying  $t$ : our algorithm recovers

<sup>4</sup> The constraints  $y_{ii} = 1, \forall i$  are satisfied when there is no outlier.



those clusters with in-cluster edge density higher than  $t$  and cross-cluster density lower than  $t$ . With a larger  $t$ , the algorithm operates at a higher resolution and detects small clusters with high density. By varying  $t$ , our algorithm can be turned into a method for *multi-resolution clustering* [1] which explores all levels of the cluster hierarchy. We leave this to future work. Importantly, the above example shows that it is generally impossible to uniquely determine the value of  $t$  from data.

### III. THE GENERALIZED STOCHASTIC BLOCKMODEL

While our algorithm above is derived as a relaxation of ML for the standard stochastic blockmodel, we establish performance guarantees in a much more general setting, which is defined by six parameters  $n_1$ ,  $n_2$ ,  $r$ ,  $K$ ,  $p$  and  $q$ ; it is described below.

**Definition 1** (Generalized Stochastic Blockmodel (GSBM)). *If  $p > q$  ( $p < q$ , resp.), consider a random graph generated as follows: The  $n = n_1 + n_2$  nodes are divided into two sets  $V_1$  and  $V_2$ . The  $n_1$  nodes in  $V_1$  are further partitioned into  $r$  disjoint sets, which we will refer to as the “true” clusters. Let  $K$  be the minimum size of a true cluster. For every pair of nodes  $i, j$  that belong to the same true cluster, edge  $(i, j)$  is present in the graph with probability that is at least (at most, resp.)  $p$ , while for every pair where the nodes are in different clusters the edge is present with probability at most (at least, resp.)  $q$ . The other  $n_2$  nodes in  $V_2$  are not in any cluster (we will call them outliers); for each  $i \in V_2$  and  $j \in V_1 \cup V_2$ , there is an edge between the pair  $i, j$  with probability at most (at least, resp.)  $q$ .*

**Definition 2** (Semi-random GSBM). *On a graph generated from GSBM with  $p > q$  ( $p < q$ , resp.), an adversary is allowed to arbitrarily (a) add (remove, resp.) edges between nodes in the same true cluster, and (b) remove (add, resp.) edges between pairs of nodes if they are in different clusters, or if at least one of them is an outlier in  $V_2$ .*

The **objective** is to find the underlying true clusters, given the graph generated from the semi-random GSBM.

The standard stochastic blockmodel/planted partition model is a special case of GSBM with  $n_2 = 0$ ,  $r \geq 2$ , all cluster sizes equal to  $K$ , and all in-cluster (cross-cluster, resp.) probabilities equal to  $p$  ( $q$ , resp.). GSBM generalizes the standard models as it allows for heterogeneity in the graph:

- $p$  and  $q$  are lower and upper bounds, instead of exact edge probabilities;
- $K$  is also a lower bound, so clusters can have different sizes;
- outliers (nodes not in any cluster) are allowed.

GSBM removes many restrictions in standard planted models and thus better models practical graphs.

The semi-random GSBM allows for further modeling power. It blends the worst case models, which are often overly pessimistic,<sup>5</sup> and the purely random graphs, which are extremely unstructured and have very special properties usually not possessed by real-world graphs [31]. This semi-random framework has been used and studied extensively in the computer science literature as a better model for real-world networks [9, 10, 11]. At first glance, the adversary seems to make the problem easier by adding in-cluster edges and removing cross-cluster edges when  $p > q$ . This is not necessarily the case. The adversary can significantly change some statistical properties of the random graph (e.g., alter spectral structure and node degrees, and create local optima by adding dense spots [10]), and foil algorithms that over-exploit such properties. In fact, some spectral algorithms that work extremely well on random models are shown to fail in the semi-random setting [8]. An algorithm that is robust against an adversary is more likely to work well on real-world graphs. As is shown later, our algorithm processes this desired property.

#### A. Special Cases

GSBM recovers as special cases many classical and widely studied models for clustered graphs, by considering different values for the parameters  $n_1$ ,  $n_2$ ,  $r$ ,  $K$ ,  $p$  and  $q$ . We classify these models into two categories based on the relation between  $p$  and  $q$ .

<sup>5</sup>For example, the minimum graph bisection problem is NP-hard.



- 1)  $p > q$ : GSBM with  $p > q$  models *homophily*, the tendency that individuals belonging to the same community tend to connect *more* than those in different communities. Special cases include:
  - *Planted Clique* [32]:  $p = 1$ ,  $r = 1$  (so  $n_1 = K$ ) and  $n_2 > 0$ ;
  - *Planted  $r$ -Clique* [5]:  $p = 1$  and  $r \geq 1$ ;
  - *Stochastic Blockmodel/Planted Partition* [4, 3]:  $n_2 = 0$ ,  $r \geq 2$  with all cluster sizes equal to  $K$ .
- 2)  $p < q$ : GSBM with  $p < q$  models *heterophily*. Special cases include:
  - *Planted Coloring* [10]:  $q > p = 0$ ,  $r \geq 2$ , and  $n_2 = 0$ ;
  - *Planted  $r$ -Cut/noisy coloring* [9, 13]:  $q > p \geq 0$ ,  $r \geq 2$ , and  $n_2 = 0$ .

In the next two sections, we describe our algorithm and provide performance guarantees under the semi-random GSBM. This implies guarantees for all the special cases above. We provide a detailed comparison with literature after presenting our results.

#### IV. MAIN RESULTS: PERFORMANCE GUARANTEES

In this section we provide analytical performance guarantees for our algorithm under the semi-random GSBM. We provide a unified theorem, and then discuss its consequences for various special cases, and compare with literature. We also discuss how to estimate the parameter  $t$  in the special case of the standard stochastic blockmodel. We shall first consider the case with  $p > q$ . The  $p < q$  case is a direct consequence and is discussed in Section IV-C3. All proofs are postponed to Sections VI to IX.

##### A. A Monotone Lemma

Our optimization-based algorithm has a nice monotone property: adding/removing edges “aligned with” the optimal  $\hat{Y}$  (as is done by the adversary) cannot result in a different optimum. This is summarized in the following lemma.

**Lemma 1.** *Suppose  $p > q$  and  $\hat{Y}$  is the unique optimum of (2)–(3) for a given  $A$ . If now we arbitrarily change some edges of  $A$  to obtain  $\tilde{A}$ , by (a) choosing some edges such that  $\hat{y}_{ij} = 1$  but  $a_{ij} = 0$ , and making  $\tilde{a}_{ij} = 1$ , and (b) choosing some edges where  $\hat{y}_{ij} = 0$  but  $a_{ij} = 1$ , and making  $\tilde{a}_{ij} = 0$ . Then,  $\hat{Y}$  is also the unique optimum of (2)–(3) with  $\tilde{A}$  as the input.*

The lemma shows that our algorithm is inherently robust under the semi-random model. In particular, the algorithm succeeds in recovering the true clusters on the semi-random GSBM as long as it succeeds on GSBM with the same parameters. In the sequel, we therefore focus solely on GSBM, with the understanding that any guarantee for it immediately implies a guarantee for the semi-random variant.

##### B. Main Theorem

Let  $Y^*$  be the matrix corresponding to the true clusters in GSBM, i.e.,  $y_{ij}^* = 1$  if and only if  $i, j \in V_1$  and they are in the same true cluster, and 0 otherwise. The theorem below establishes conditions under which our algorithm, specifically the convex program (2)–(3), yields this  $Y^*$  as the unique optimum (without any further need for rounding etc.) with high probability (w.h.p.). Throughout the paper, *with high probability* means with probability at least  $1 - c_0 n^{-8}$  for some universal absolute constant  $c_0$ .

**Theorem 1.** *Suppose the graph  $A$  is generated according to GSBM with  $p > q$ . If  $t$  in (4) is chosen to satisfy*

$$\frac{1}{4}p + \frac{3}{4}q \leq t \leq \frac{3}{4}p + \frac{1}{4}q, \quad (5)$$

*then  $Y^*$  is the unique optimal solution to the convex program (2)–(3) w.h.p. provided*

$$\frac{p - q}{\sqrt{p(1 - q)}} \geq c_1 \max \left\{ \frac{\sqrt{n}}{K}, \frac{\log^2 n}{\sqrt{K}} \right\}, \quad (6)$$

where  $c_1$  is an absolute constant independent of  $p, q, K$  and  $n$ .

Our theorem quantifies the tradeoff between the four parameters governing the hardness of GSBM— the minimum in-cluster edge density  $p$ , the maximum across-cluster edge density  $q$ , the minimum cluster size  $K$  and the number of outliers  $n_2 = n - n_1$ — required for our algorithm to succeed, i.e., to recover the underlying true clustering without any error. Note that we can handle any values of  $p, q, n_2$  and  $K$  as long as they satisfy the condition in the theorem; in particular, they are allowed to scale with  $n$ . Interestingly, the theorem does not have an explicit dependence on the number of clusters  $r$  except via the relation  $rK \leq n$ .

We now discuss the *tightness* of Theorem 1 in terms of these parameters. When  $K = \Theta(n)$ , we have a matching converse result.

**Theorem 2.** *Suppose all clusters have equal size  $K$ , and the in-cluster (cross-cluster, resp.) edge probabilities are uniformly  $p$  ( $q$ , resp.), with  $K = \Theta(n)$  and  $n_2 = \Theta(n_1)$ . Under GSBM with  $p > q$  and  $n$  sufficiently large, for any algorithm to correctly recover the clusters with probability at least  $\frac{3}{4}$ , we must have*

$$\frac{p - q}{\sqrt{p(1 - q)}} \geq c_2 \frac{1}{\sqrt{n}},$$

$c_2$  is an absolute constant.

This theorem gives a necessary condition for *any* algorithm to succeed regardless of its computational complexity. It shows that Theorem 1 is optimal up to logarithmic factors for all values of  $p$  and  $q$  when  $K = \Theta(n)$ .

For smaller values of  $K$ , notice that Theorem 1 requires  $K$  to be  $\Omega(\sqrt{n})$ , since the left hand side of (6) is less than 1. This lower-bound is achieved when  $p$  and  $q$  are both constants independent of  $n$  and  $K$ . There are reasons to believe that this requirement is unlikely to be improvable using polynomial-time algorithms. Indeed, the special case with  $p = 1$  and  $q = \frac{1}{2}$  corresponds to the classical planted clique problem [32]; finding a clique of size  $K = o(\sqrt{n})$  is widely believed to be computationally hard even on average and has been used as a hard problem for cryptographic applications [33, 34].

For other values of  $p$  and  $q$ , no general and rigorous converse result exists. However, there are evidences suggesting that no other polynomial-time algorithm is likely to have better guarantees than our result (6). The authors of [13] show, using non-rigorous but deep arguments from statistical physics, that recovering the clustering is impossible in polynomial time if  $\frac{p-q}{\sqrt{p}} = o\left(\frac{\sqrt{n}}{K}\right)$ . Moreover, the work in [14] shows that a large class of spectral algorithms fail under the same condition. In view of these results, it is possible that our algorithm is optimal w.r.t. all polynomial-time algorithms for all values of  $p, q$  and  $K$ .

Several further remarks regarding Theorem 1 are in order.

- A nice feature of our result is that we only need  $p - q$  to be large *only as compared to*  $\sqrt{p}$ ; several other existing results (see Table I) require a lower bound (as a function of  $n$  and  $K$ ) on  $p - q$  itself. When  $K$  is  $\Theta(n)$ , we allow  $p$  and  $p - q$  to be as small as  $\Theta(\log^4(n)/n)$ .
- The number of clusters  $r$  is allowed to grow rapidly with  $n$ ; this is called the high-dimensional setting [21]. In particular, our algorithm can recover as many as  $r = \Theta(\sqrt{n})$  equal size clusters. Any algorithm with a better scaling would recover cliques of size  $o(\sqrt{n})$ , an unlikely task in polynomial time in light of the hardness of the planted clique problem discussed above.
- The number of outliers can also be large, as many as  $n_2 = \Theta(n) = \Theta(n_1^2)$ , which is attained when  $p - q, r$  are  $\Theta(1)$  and  $K = \Theta(\sqrt{n_2})$ . In other words, almost all the nodes can be outliers, and this is true even when there are multiple clusters that are not cliques (i.e.,  $p < 1$ ).
- Not all existing methods can handle non-uniform edge probabilities and node degrees, which often require special treatment (see [12]). This issue is addressed seamlessly by our method by definition of GSBM.

In the following sub-section, we discuss various planted problems to which Theorem 1 applies and compare with existing work. Our results match the best existing results in all cases (up to logarithm factors), and in many important settings lead to order-wise stronger guarantees.

### C. Consequences and Comparison with Literature

1) *Standard Stochastic Blockmodel (a.k.a. Planted Partition Model)*: This model assumes that all clusters have the same size  $K$  with no isolated nodes ( $n_2 = 0$ ), and the in-cluster and across-cluster edge probabilities are uniformly  $p$  and  $q$ , respectively, with  $p > q$ . We compare our result to past approaches and theoretical results in Table I. Our result has the scaling  $p - q = \Omega\left(\frac{\sqrt{pn}}{K}\right)$  and  $p = \Omega\left(\frac{n}{K^2}\right)$ , which improves on all existing results by polynomial factors. This means that we can handle much noisier and sparser graphs, especially when the number of clusters  $r = n/K$  is growing. A recent paper [35], which appeared after the publication of the conference version [36] of this manuscript, proposes a tensor approach for graph clustering. Our guarantee is a few logarithmic factors better than their results applied to the standard stochastic blockmodel.

2) *Planted  $r$ -Clique Problem*: Here the task is to find a set of  $r$  disjoint cliques, each of size at least  $K$ , that have been planted in an Erdos-Renyi random graphs  $G(n, q)$ . Setting  $p = 1$  in Theorem 1, we obtain the following guarantee for the planted  $r$ -clique problem.

**Corollary 1.** *For the planted  $r$ -clique problem, the formulation (2)-(4) with  $t$  chosen according to Theorem 1 finds the hidden cliques w.h.p. provided*

$$1 - q \geq c_3 \max \left\{ \frac{n}{K^2}, \frac{\log^4 n}{K} \right\},$$

where  $c_3$  is an absolute constant.

In the regime where  $r$  is allowed to scale with  $n$  and  $q$  bounded away from zero, the best previous results are given in [5] ( $1 - q = \Omega(\frac{rn}{K^2})$ ) and in [23] ( $1 - q = \Omega(\frac{\sqrt{n}}{K})$ ). Corollary 1 is stronger than both of them for large  $r$ .

3) *The Heterophily Case ( $p < q$ )*: Given a graph  $A$  generated from the semi-random GSBM with intra/inter-cluster densities  $p < q$ , we can run our algorithm to the graph  $A' = \mathbf{1}\mathbf{1}^\top - A$ , where  $\mathbf{1}\mathbf{1}^\top$  is the all-one matrix. Note that  $A'$  can be considered as generated from GSBM with intra/inter-cluster densities  $p' = 1 - p$  and  $q' = 1 - q$ , where  $p' > q'$ . With this reduction, Lemma 1 and Theorem 1 immediately yield the following guarantee.

**Corollary 2.** *Under the semi-random GSBM with  $p < q$ , the formulation (2)-(4) applied to  $\mathbf{1}\mathbf{1}^\top - A$  with  $t$  obeying*

$$\frac{3}{4}p + \frac{1}{4}q \leq 1 - t \leq \frac{1}{4}p + \frac{3}{4}q$$

*finds the true clustering w.h.p. provided*

$$q - p \geq c_3 \sqrt{(1 - p)q} \max \left\{ \frac{\sqrt{n}}{K}, \frac{\log^2 n}{\sqrt{K}} \right\},$$

where  $c_3$  is an absolute constant.

This corollary immediately yields guarantees for the planted coloring problem [7] and the planted  $r$ -cut [9] (a.k.a. planted noisy coloring [13]) problem. We are not aware of any exiting work that explicitly considers the mirrored GSBM in its general form ( $n_2 > 0$ ,  $1 \geq q > p \geq 0$ , and  $K = O(n)$  with potential non-random edges). However, since any guarantee for GSBM with  $p > q$  implies a guarantee for GSBM with  $p < q$ , Table I provides a comparison with existing work when  $n_2 = 0$  and the edge probabilities and cluster sizes are uniform. Again our guarantee outperforms all existing ones.

4) *Planted Coloring Problems*: A special case of the above problem is the planted coloring problem, where  $p = 0$  and  $n_2 = 0$ . The best existing result  $q = \Omega\left(\frac{n}{K^2} \vee \frac{\log n}{K}\right)$  is given by various algorithms (e.g., [7, 8]). By Corollary 2, our algorithm succeeds provided  $q = \Omega\left(\frac{n}{K^2} \vee \frac{\log^4 n}{K}\right)$ . We match the best existing algorithms for  $K = O(n/\log^4(n))$ , and are off by a few log factors for larger  $K$ .

#### D. Estimating $t$ in Special Cases

We have argued that specifying  $t$  in a completely data-driven way is ill-posed for the general GSBM, e.g., when the clusters are hierarchical. However, for special cases this can be done reliably with strong guarantees. Consider the standard stochastic blockmodel, where all clusters have the same size  $K$ , the edge probabilities are uniform (i.e., equal to  $p$  within clusters and  $q$  across clusters, with  $p > q$ ), and there are no outliers ( $n_2 = 0$ ) or non-random edges. Observe that  $\mathbb{E}[A] - (1-p)I$  is a matrix with blocks of  $p$  and  $q$ 's<sup>6</sup>, which is equal to the Kronecker product of a  $K \times K$  all-one matrix and an  $r \times r$  circulant matrix with entries equal to  $p$  on the diagonal and  $q$  elsewhere. The all-one matrix has one non-zero eigenvalue  $K$ , and the circulant matrix has eigenvalues  $(p-q) + rq$  and  $p-q$  with multiplicities 1 and  $r-1$ , respectively. The eigenvalues of  $\mathbb{E}[A] - (1-p)I$  is the product of these two matrices. It follows that the first eigenvalue of  $\mathbb{E}[A]$  is  $K(p-q) + nq + (1-p)$  with multiplicity 1, the second eigenvalue is  $K(p-q) + (1-p)$  with multiplicity  $\frac{n}{K} - 1$ , and the third eigenvalue is  $1-p$  with multiplicities  $(n - \frac{n}{K})$  [18]. This motivates us to use the eigenvalues of the observed matrix  $A$  to estimate  $p$ ,  $q$  and  $t$ ; see Algorithm 2.

---

#### Algorithm 2 Estimation of $t$ from data

---

- 1) Compute and sort the eigenvalues of  $A$ , denoted as  $\hat{\lambda}_1 > \hat{\lambda}_2 > \dots > \hat{\lambda}_n$ .
- 2) Let  $\hat{r} = \arg \max_{i=2, \dots, n-1} (\hat{\lambda}_i - \hat{\lambda}_{i+1})$ . Set  $\hat{K} = n/\hat{r}$ .
- 3) Set

$$\begin{cases} \hat{p} = \frac{\hat{K}\hat{\lambda}_1 + (n-\hat{K})\hat{\lambda}_2 - n}{n(\hat{K}-1)}, \\ \hat{q} = \frac{\hat{\lambda}_1 - \hat{\lambda}_2}{n}. \end{cases}$$

- 4) Set  $t = \frac{\hat{p} + \hat{q}}{2}$ .
- 

The following theorem guarantees that the estimation errors are small.

**Theorem 3.** *Under the standard stochastic blockmodel and the condition (6) in Theorem 1, the parameters estimated in Algorithm 2 satisfy the following with high probability, where  $c_4$  is an absolute positive constant:*

$$\begin{aligned} \hat{K} &= K, \\ \max\{|\hat{p} - p|, |\hat{q} - q|\} &\leq c_4 \frac{\sqrt{p(1-q)n}}{K}, \\ \frac{1}{4}p + \frac{3}{4}q &\leq t \leq \frac{3}{4}p + \frac{1}{4}q. \end{aligned}$$

In particular, the estimated  $t$  satisfies the condition (5) in Theorem 1. The above theorem also ensures that Algorithm 2 is a consistent estimator of the parameters  $p$  and  $q$  when condition (6) is satisfied, a result of independent interest. Combining Theorem 1 and Theorem 3, we obtain a complete algorithm that is guaranteed to find the clusters under the standard stochastic blockmodel obeying condition (6), without knowledge of any generative parameters of the underlying model.

<sup>6</sup>Recall that we use the convention  $a_{ii} = 1$ .

## V. EMPIRICAL RESULTS

### A. Implementation Issues

The convex program (2)–(3) can be solved using a general purpose SDP solver, but this method does not scale well to problems with more than a few hundred nodes. To facilitate fast and efficient solution, we propose to use a family of first-order algorithms called Augmented Lagrange Multiplier (ALM) methods. Note that the program (2)–(3) can be rewritten as

$$\begin{aligned} \min_{Y, S \in \mathbb{R}^{n \times n}} \quad & \lambda \|C \circ S\|_1 + \|Y\|_* \\ \text{s.t} \quad & Y + S = A, \\ & 0 \leq y_{ij} \leq 1, \forall i, j, \end{aligned} \tag{7}$$

where  $\lambda := \frac{1}{48\sqrt{n}}$ , the matrix  $C \in \mathbb{R}^{n \times n}$  satisfies  $c_{ij} = c_A$  if  $a_{ij} = 1$  and  $c_{ij} = c_{A^c}$  otherwise, and  $\circ$  denotes the element-wise matrix product. This problem can be recognized as a generalization of the standard convex formulation of the low-rank and sparse matrix decomposition problem [25, 24], of which the numerical solution has been well studied. We adapt the ALM method in [37] to the above problem, given in Algorithm 3. Here  $\mathcal{S}_{\epsilon C}(\cdot) : \mathbb{R}^{n \times n} \mapsto \mathbb{R}^{n \times n}$  is the element-wise weighted soft-thresholding operator,

---

**Algorithm 3** ALM for Minimizing Nuclear Norm plus Weighted  $\ell_1$  Norm

---

Input:  $A, C \in \mathbb{R}^{n \times n}$ .

Initialize:  $M^{(0)} = 0$ ;  $Y^{(0)} = 0$ ;  $S^{(0)} = 0$ ;  $\mu_0 > 0$ ;  $\alpha > 1$ ;  $k = 0$ ,  $\lambda = \frac{1}{48\sqrt{n}}$ .

**while** not converge **do**

$(U, \Sigma, V) = \text{svd}(A - S^{(k)} + \mu_k^{-1} M^{(k)})$ .

$\bar{Y}^{(k+1)} = U \mathcal{S}_{\mu_k^{-1}}(\Sigma) V$ .

    For all  $(i, j)$ ,  $y_{ij}^{(k+1)} = \max \left\{ \min \left\{ \bar{Y}_{ij}^{(k+1)}, 1 \right\}, 0 \right\}$ .

$S^{(k+1)} = \mathcal{S}_{\mu_k^{-2} \lambda C}(A - Y^{(k+1)} + \mu_k^{-1} M^{(k)})$ .

$M^{(k+1)} = M^{(k)} + \mu_k (A - Y^{(k+1)} - S^{(k+1)})$ .

$\mu_{k+1} = \alpha \mu_k$ ,  $k = k + 1$ .

**end while**

Return  $Y^{(k+1)}, S^{(k+1)}$ .

---

defined as

$$(\mathcal{S}_{\epsilon C}(X))_{ij} = \begin{cases} x_{ij} - \epsilon c_{ij}, & \text{if } x_{ij} > \epsilon c_{ij} \\ x_{ij} + \epsilon c_{ij}, & \text{if } x_{ij} < -\epsilon c_{ij} \\ 0, & \text{otherwise.} \end{cases}$$

In other words, it shrinks each entry of  $X$  towards zero by  $\epsilon c_{ij}$ . The unweighted version  $\mathcal{S}_{\epsilon}(\cdot) \triangleq \mathcal{S}_{\epsilon I}(\cdot)$  is also used. The stopping criteria and parameters of the algorithm are chosen similarly to [37].

### B. Simulations

We perform experiments on synthetic data, and compare with other methods. We generate a graph using the stochastic blockmodel with  $n = 1000$  nodes,  $r = 5$  clusters with equal size  $K = 200$ , and  $p, q \in [0, 1]$ . We apply our method to the graph, where we pick  $t$  using Algorithm 2 and solve the optimization problem using Algorithm 3. Due to numerical accuracy, the output  $\hat{Y}$  of our algorithm may not be strictly integer, so we do the following simple rounding: compute the mean  $\bar{y}$  of the entries of  $\hat{Y}$ , and round each entry of  $\hat{Y}$  to 1 if it is greater than  $\bar{y}$ , and 0 otherwise. We measure the error by  $\|Y^* - \text{round}(\hat{Y})\|_1$ , which equals the number of misclassified pairs. We say our method succeeds if it misclassifies less than 0.1% of the pairs.

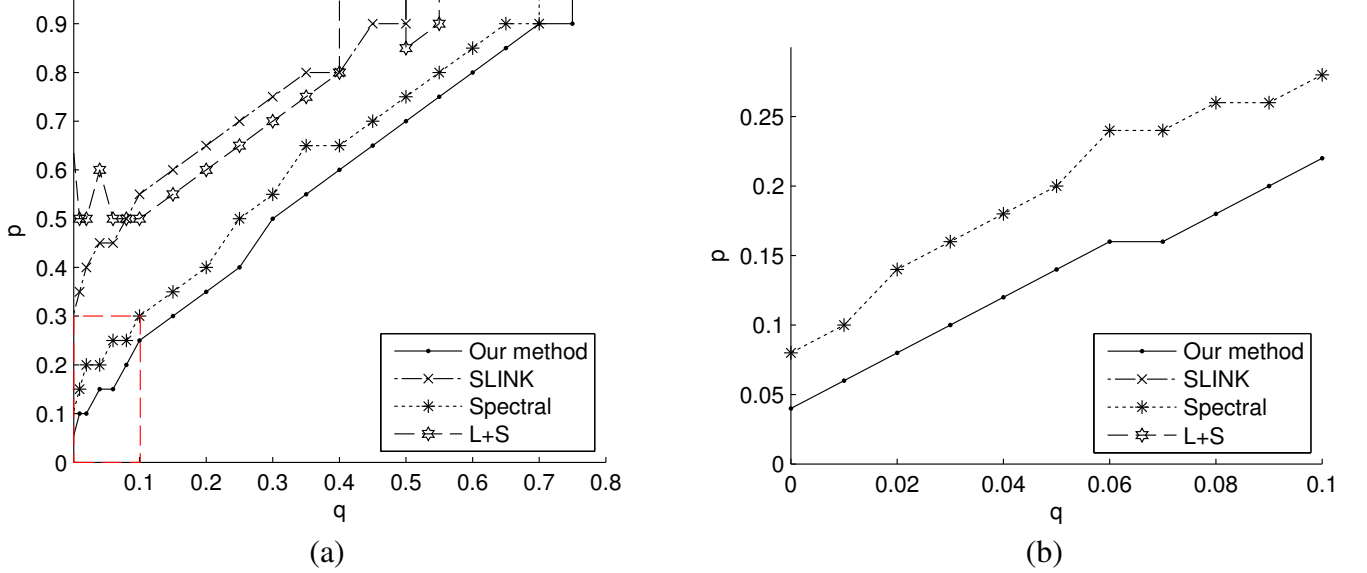


Fig. 1. (a) Comparison of our method with Single-Linkage clustering (SLINK), spectral clustering, and low-rank-plus-sparse (L+S) approach. The area above each curve is the values of  $(p, q)$  for which a method successfully recovers the underlying true clusters. (b) More detailed results for the area in the box in (a). The experiments are conducted on synthetic data with  $n = 1000$  nodes and  $r = 5$  clusters with equal size  $K = 200$ .

For comparison, we consider three alternative methods: (1) Single-Linkage clustering (SLINK) [38], which is a hierarchical clustering method that merges the most similar clusters in each iteration. We use the difference of neighbors, namely  $\|A_i - A_j\|_1$ , as the distance measure of nodes  $i$  and  $j$ , and terminate when SLINK finds a clustering with  $r = 5$  clusters. (2) A spectral clustering method [39], where we run SLINK on the top  $r = 5$  singular vectors of  $A$ . (3) The low-rank-plus-sparse approach [28, 22], followed by the rounding scheme described in the last paragraph. Note the first two methods assume knowledge of the number of clusters  $r$ , which is not available to our method.

For each  $q$ , we find the smallest  $p$  for which a method succeeds, and average over 20 trials. The results are shown in Figure 1(a), where the area above each curves corresponds to the range of feasible  $(p, q)$  for each method. It can be seen that our method subsumes all others, in that we succeed for a strictly larger range of  $(p, q)$ . Figure 1(b) shows more detailed results for sparse graphs ( $p \leq 0.3, q \leq 0.1$ ), for which SLINK and low-rank-plus-sparse approach completely fail, while our method significantly outperforms the spectral method, the only alternative method that works in this regime.

## VI. PROOF OF LEMMA 1

In this section we prove the monotone lemma. Set  $\lambda = \frac{1}{48\sqrt{n}}$ . Define  $\Omega_+ = \{(i, j) : a_{ij} = 0, \tilde{a}_{ij} = 1\}$  and  $\Omega_- = \{(i, j) : a_{ij} = 1, \tilde{a}_{ij} = 0\}$ . Let  $Y \neq \hat{Y}$  be an arbitrary alternative feasible solution obeying  $0 \leq y_{ij} \leq 1, \forall i, j$ . By optimality of  $\hat{Y}$  to the original program, we have

$$\left( c_{\mathcal{A}} \sum_{a_{ij}=1} \hat{y}_{ij} - c_{\mathcal{A}^c} \sum_{a_{ij}=0} \hat{y}_{ij} \right) - \frac{1}{\lambda} \|\hat{Y}\|_* > \left( c_{\mathcal{A}} \sum_{a_{ij}=1} y_{ij} - c_{\mathcal{A}^c} \sum_{a_{ij}=0} y_{ij} \right) - \frac{1}{\lambda} \|Y\|_*.$$

Next, by definition of  $\tilde{A}$ ,  $\Omega_+$  and  $\Omega_-$ , we have

$$\left( c_{\mathcal{A}} \sum_{\tilde{a}_{ij}=1} \hat{y}_{ij} - c_{\mathcal{A}^c} \sum_{\tilde{a}_{ij}=0} \hat{y}_{ij} \right) - \left( c_{\mathcal{A}} \sum_{a_{ij}=1} \hat{y}_{ij} - c_{\mathcal{A}^c} \sum_{a_{ij}=0} \hat{y}_{ij} \right) = \sum_{(i,j) \in \Omega_+} (c_{\mathcal{A}} + c_{\mathcal{A}^c});$$



and

$$\begin{aligned}
& \left( c_{\mathcal{A}} \sum_{a_{ij}=1} y_{ij} - c_{\mathcal{A}^c} \sum_{a_{ij}=0} y_{ij} \right) - \left( c_{\mathcal{A}} \sum_{\tilde{a}_{ij}=1} y_{ij} - c_{\mathcal{A}^c} \sum_{\tilde{a}_{ij}=0} y_{ij} \right) \\
&= (c_{\mathcal{A}} + c_{\mathcal{A}^c}) \sum_{(i,j) \in \Omega_-} y_{ij} - (c_{\mathcal{A}} + c_{\mathcal{A}^c}) \sum_{(i,j) \in \Omega_+} y_{ij} \\
&\geq - \sum_{(i,j) \in \Omega_+} (c_{\mathcal{A}} + c_{\mathcal{A}^c}),
\end{aligned}$$

where we use  $0 \leq y_{ij} \leq 1$  for all  $(i, j)$  in the last inequality. Summing the L.H.S. and R.H.S. of the last three display equations establishes that

$$\left( c_{\mathcal{A}} \sum_{\tilde{a}_{ij}=1} \hat{y}_{ij} - c_{\mathcal{A}^c} \sum_{\tilde{a}_{ij}=0} \hat{y}_{ij} \right) - \frac{1}{\lambda} \|\hat{Y}\|_* > \left( c_{\mathcal{A}} \sum_{\tilde{a}_{ij}=1} y_{ij} - c_{\mathcal{A}^c} \sum_{\tilde{a}_{ij}=0} y_{ij} \right) - \frac{1}{\lambda} \|Y\|_*.$$

Since  $Y$  is arbitrary, we conclude that  $\hat{Y}$  is the unique optimum of the modified program.

## VII. PROOF OF THEOREM 1

We prove our main theorem in this section. The proof consists of three main steps, which we elaborate below.

### A. Step 1: Reduction to Homogeneous Edge Probabilities

We show that it suffices to assume that the in-cluster edge probability is uniformly  $p$ , and the across-cluster edge probability is uniformly  $q$ . In the heterogeneous model, suppose an edge is placed between nodes  $i$  and  $j$  with probability  $p_{ij}$  if they are in the same cluster, where  $p_{ij} \geq p$ . This is equivalent to the following two-step model: first flip a coin with head probability  $p$ , and add the edge if it is head; if it is tail, then flip another coin and add the edge with probability  $\frac{p_{ij}-p}{1-p}$ . By the monotone property in Lemma 1, we know that if our convex program succeeds on the graph generated in the first step, then it also succeeds for the second step, because more in-cluster edges are added. A similar argument applies to the across-cluster edges. Therefore, heterogeneous edge probabilities only make the probability of success higher, and thus we only need to prove the homogeneous case.

### B. Step 2: Optimality Condition

We need some additional notation. We denote the singular value decomposition of  $Y^*$  (notice  $Y^*$  is symmetric and positive definite) by  $U_0 \Sigma_0 U_0^\top$ . For any matrix  $M$ , we define  $P_T(M) := U_0 U_0^\top M + M U_0 U_0^\top - U_0 U_0^\top M U_0 U_0^\top$ . For a set  $\Omega$  of matrix indices, let  $P_\Omega(M)$  be the matrix obtained by setting the entries of  $M$  outside  $\Omega$  to zero, and we use  $\sum_\Omega$  as a shorthand of  $\sum_{(i,j) \in \Omega}$ . Define the sets  $\mathcal{A} := \text{support}(A)$  and  $R := \text{support}(Y^*) = \text{support}(U_0 U_0^\top)$ . The true cluster matrix  $Y^*$  is an optimal solution to the program (2)–(3) if

$$\lambda c_{\mathcal{A}} \sum_{\mathcal{A}} (y_{ij}^* - y_{ij}) - \lambda c_{\mathcal{A}^c} \sum_{\mathcal{A}^c} (y_{ij}^* - y_{ij}) - (\|Y^*\|_* - \|Y\|_*) \geq 0 \quad (8)$$

for all feasible  $Y$  obeying (3). Suppose there is a matrix  $W$  that satisfies

$$\|W\| \leq 1, P_T(W) = 0. \quad (9)$$



The matrix  $U_0 U_0^\top + W$  is a subgradient of  $f(X) = \|X\|_*$  at  $X = Y^*$ , so  $\|Y\|_* - \|Y^*\|_* \geq \langle U_0 U_0^\top + W, Y - Y^* \rangle$ . Therefore, (8) is implied by

$$\lambda c_{\mathcal{A}} \sum_{\mathcal{A}} (y_{ij}^* - y_{ij}) - \lambda c_{\mathcal{A}^c} \sum_{\mathcal{A}^c} (y_{ij}^* - y_{ij}) + \langle U_0 U_0^\top + W, Y - Y^* \rangle \geq 0, \quad \forall 0 \leq Y \leq 1. \quad (10)$$

The above inequality holds in particular for any feasible  $Y$  of the form  $Y = Y^* - e_i e_j^\top$  with  $(i, j) \in R$  or  $Y = Y^* + e_i e_j^\top$  with  $(i, j) \in R^c$ . This leads to the following element-wise inequalities:

$$\begin{aligned} -\lambda c_{\mathcal{A}^c} - (U_0 U_0^\top + W)_{ij} &\geq 0, & \forall (i, j) \in R \cap \mathcal{A}^c, \\ -\lambda c_{\mathcal{A}} + w_{ij} &\geq 0, & \forall (i, j) \in R^c \cap \mathcal{A}, \\ \lambda c_{\mathcal{A}} - (U_0 U_0^\top + W)_{ij} &\geq 0, & \forall (i, j) \in R \cap \mathcal{A}, \\ \lambda c_{\mathcal{A}^c} + w_{ij} &\geq 0, & \forall (i, j) \in R^c \cap \mathcal{A}^c. \end{aligned} \quad (11)$$

It is easy to see that these inequalities are actually equivalent to (10), so together with (9) they form a sufficient condition for the optimality of  $Y^*$ .

Finding a “dual certificate”  $W$  obeying the exact conditions (9) and (11) is difficult, and does not guarantee uniqueness of the optimum. Instead, we consider an alternative sufficient condition that only requires a  $W$  that *approximately* satisfies the exact conditions. This is done in Proposition 1 below (proved in Section VII-D), which significantly simplifies the construction of  $W$ . Note that condition (b) in the proposition is a relaxation of the equality in (9), whereas condition (c) tightens (11). Setting  $\epsilon = 0$  and changing equalities to inequalities in the proposition recover the exact conditions.

**Proposition 1.**  $Y^*$  is the unique optimal solution to the program (2)–(3), if there exists a matrix  $W \in \mathbb{R}^{n \times n}$  and a number  $0 < \epsilon < 1$  that satisfy the following conditions: (a)  $\|W\| \leq 1$ , (b)  $\|P_T(W)\|_\infty \leq \frac{\epsilon}{2} \lambda \min\{c_{\mathcal{A}^c}, c_{\mathcal{A}}\}$ , and (c)

$$\begin{aligned} -(1 + \epsilon) \lambda c_{\mathcal{A}^c} - (U_0 U_0^\top + W)_{ij} &= 0, & \forall (i, j) \in R \cap \mathcal{A}^c, \\ -(1 + \epsilon) \lambda c_{\mathcal{A}} + w_{ij} &= 0, & \forall (i, j) \in R^c \cap \mathcal{A}, \\ (1 - \epsilon) \lambda c_{\mathcal{A}} - (U_0 U_0^\top + W)_{ij} &\geq 0, & \forall (i, j) \in R \cap \mathcal{A}, \\ (1 - \epsilon) \lambda c_{\mathcal{A}^c} + w_{ij} &\geq 0, & \forall (i, j) \in R^c \cap \mathcal{A}^c. \end{aligned}$$

### C. Step 3: Constructing $W$

We build a  $W$  that satisfies the conditions in Proposition 1 w.h.p. We use  $\mathbf{1}$  to denote the all-one column vector in  $\mathbb{R}^n$ , so  $\mathbf{1}\mathbf{1}^\top$  is the all-one  $n \times n$  matrix. Let  $E = \{(i, i), i = 1, \dots, n\}$  be the set of diagonal entries. For an  $\epsilon$  to be specified later, we define  $W = W_1 + W_2 + W_3 + W_4$  with  $W_i$  given by

$$\begin{aligned} W_1 &= -P_{R \cap \mathcal{A}^c}(U_0 U_0^\top) + \frac{1-p}{p} P_{R \cap \mathcal{A}}(U_0 U_0^\top), \\ W_2 &= (1 + \epsilon) \lambda c_{\mathcal{A}^c} \left[ -P_{R \cap \mathcal{A}^c}(\mathbf{1}\mathbf{1}^\top) + \frac{1-p}{p} P_{R \cap \mathcal{A}}(\mathbf{1}\mathbf{1}^\top) \right], \\ W_3 &= (1 + \epsilon) \lambda c_{\mathcal{A}} \left[ P_{(R^c \cap E^c) \cap \mathcal{A}}(\mathbf{1}\mathbf{1}^\top) - \frac{q}{1-q} P_{(R^c \cap E^c) \cap \mathcal{A}^c}(\mathbf{1}\mathbf{1}^\top) \right], \\ W_4 &= (1 + \epsilon) \lambda c_{\mathcal{A}} P_{R^c}(I), \end{aligned}$$

where  $I$  is the identity matrix. We briefly explain the ideas behind the construction. Each of the matrices  $W_1$ ,  $W_2$  and  $W_3$  is the sum of two terms. The first term is derived from the equalities in condition (c) in Proposition 1. The second term is constructed in such a way that each  $W_i$  is a zero-mean random matrix (due to the randomness in  $\mathcal{A}$ ), so it is likely to have small norms and satisfy conditions (a) and (b). The matrix  $W_4$  accounts for the outlier nodes. In particular, it is a diagonal matrix with  $(W_4)_{ii}$  being non-zero if and only if  $i \in V_2$

The following proposition (proved in Section VII-E) shows that  $W$  indeed satisfies all the desired conditions w.h.p., hence establishing Theorem 1.

**Proposition 2.** *Under the conditions in Theorem 1,  $W$  constructed above satisfies the conditions (a)–(c) in Proposition 1 w.h.p. with*

$$\epsilon := \frac{48}{\sqrt{t(1-t)}} \max \left\{ \frac{\sqrt{n}}{K}, \sqrt{\frac{\log^4 n}{K}} \right\}.$$

#### D. Proof of Proposition 1 (Optimality Condition)

Let  $P_{T^\perp}(W) := W - P_T(W)$ . Consider any feasible solution  $Y$  and let  $\Delta := Y - Y^*$ . The difference between the objective values of  $Y$  and  $Y^*$  satisfies

$$\begin{aligned} (*) &:= c_{\mathcal{A}} \sum_{\mathcal{A}} \delta_{ij} - c_{\mathcal{A}^c} \sum_{\mathcal{A}^c} \delta_{ij} - \frac{1}{\lambda} \|Y^* + \Delta\|_* + \frac{1}{\lambda} \|Y^*\|_* \\ &\leq c_{\mathcal{A}} \sum_{\mathcal{A}} \delta_{ij} - c_{\mathcal{A}^c} \sum_{\mathcal{A}^c} \delta_{ij} - \frac{1}{\lambda} \langle U_0 U_0^\top + P_{T^\perp}(W), \Delta \rangle \\ &= c_{\mathcal{A}} \sum_{\mathcal{A}} \delta_{ij} - c_{\mathcal{A}^c} \sum_{\mathcal{A}^c} \delta_{ij} - \frac{1}{\lambda} \langle U_0 U_0^\top + W, \Delta \rangle + \frac{1}{\lambda} \langle P_T W, \Delta \rangle, \end{aligned}$$

where in the inequality we use the fact that  $U_0 U_0^\top + P_{T^\perp}(W)$  is a subgradient of  $\|Y\|_*$  at  $Y^*$ , a consequence of condition (a) in the proposition and  $\|P_{T^\perp}(W)\| \leq \|W\|$ . We substitute condition (c) into the third term in the R.H.S. of the last display equation to obtain

$$\begin{aligned} (*) &\leq \epsilon c_{\mathcal{A}} \sum_{R \cap \mathcal{A}} \delta_{ij} - \epsilon c_{\mathcal{A}^c} \sum_{R^c \cap \mathcal{A}^c} \delta_{ij} + \epsilon c_{\mathcal{A}^c} \sum_{R \cap \mathcal{A}^c} \delta_{ij} - \epsilon c_{\mathcal{A}} \sum_{R^c \cap \mathcal{A}} \delta_{ij} + \frac{1}{\lambda} \langle P_T W, \Delta \rangle \\ &\leq -\epsilon \min\{c_{\mathcal{A}}, c_{\mathcal{A}^c}\} \|\Delta\|_1 + \frac{1}{\lambda} \langle P_T W, \Delta \rangle, \end{aligned}$$

where we use the fact that  $\delta_{ij} \leq 0$  for  $(i, j) \in R$  and  $\delta_{ij} \geq 0$  for  $(i, j) \in R^c$  since  $Y = Y^* + \Delta$  satisfies (3). Applying condition (b) yields

$$(*) \leq -\epsilon \min\{c_{\mathcal{A}^c}, c_{\mathcal{A}}\} \|\Delta\|_1 + \frac{1}{\lambda} \|P_T W\|_\infty \|\Delta\|_1 \leq -\frac{\epsilon}{2} \min\{c_{\mathcal{A}^c}, c_{\mathcal{A}}\} \|\Delta\|_1.$$

The last R.H.S. is strictly negative whenever  $\Delta \neq 0$ . This proves that  $Y^*$  is the unique optimal solution.

#### E. Proof of Proposition 2

We show that  $W$  constructed in Section VII-C satisfies the conditions in Proposition 1 w.h.p. We need two technical lemmas. First notice that the conditions (5) and (6) in Theorem 1 imply bounds on various quantities.

**Lemma 2.** *Under conditions (5) and (6) in Theorem 1, we have (1)  $p(1-q) \geq t(1-t) \geq c \max\left\{\frac{n}{K^2}, \frac{\log^4 n}{K}\right\}$ , and (2)  $\epsilon < \frac{1}{2}$ .*

*Proof.* Since  $1 > t > 0$ , we have  $t(1-t) \geq \frac{1}{2} \min\{t, 1-t\}$ . Under condition (5) on  $t$ , we further have  $\min\{t, 1-t\} \geq \frac{1}{4}(p-q)$  and  $p(1-q) \geq t(1-t)$ . It then follows from condition (6) that

$$t(1-t) \geq \frac{1}{8}(p-q) \geq c' \sqrt{t(1-t)} \max \left\{ \frac{\sqrt{n}}{K}, \sqrt{\frac{\log^4 n}{K}} \right\},$$

which implies the inequalities in part (1) of the lemma. Part (2) follows directly from part (1) and the definition of  $\epsilon$ .  $\square$

Due to the randomness of  $\mathcal{A}$ ,  $W_1$ ,  $W_2$  and  $W_3$  are symmetric random matrices with independent zero-mean entries. The support and variance of their entries are bounded in the following lemma.

**Lemma 3.** *The following holds under the conditions in Theorem 1.*

- 1) For  $i = 1, 2, 3$ , the absolute values of the entries of  $W_i$  are bounded by  $B_i$  a.s. and their variance is bounded by  $\sigma_i^2$ , where

$$\begin{aligned} B_1 &= \frac{1}{pK}, & \sigma_1^2 &= \frac{1}{pK^2}, \\ B_2 &= \frac{2}{p}\lambda_{\mathcal{C}_{\mathcal{A}^c}}, & \sigma_2^2 &= \frac{4(1-t)}{p}\lambda^2 c_{\mathcal{A}^c}^2, \\ B_3 &= \frac{2}{1-q}\lambda_{\mathcal{C}_{\mathcal{A}}}, & \sigma_3^2 &= \frac{4t}{1-q}\lambda^2 c_{\mathcal{A}}^2. \end{aligned}$$

- 2) We have  $\sigma_i \geq c \frac{B_i \log^2 n}{\sqrt{K}}$  for  $i = 1, 2, 3$ .

*Proof.* The first part of the lemma follows from the definitions of the  $W_i$ 's,  $q \leq t \leq p$  and  $\epsilon < \frac{1}{2}$  (Lemma 2). The second part follows from part (1) of Lemma 2.  $\square$

We now proceed with the proof of Proposition 2. The proof has three sub-steps, corresponding to checking the three conditions in Proposition 1.

**(1) Bounding  $\|W\|$ .**

Recall that  $W_1$  is a random matrix with i.i.d. entries, and their absolute values and variance are bounded in Lemma 3. We apply standard bounds on the spectral norm of random matrices (Lemma 4 in the Appendix) to obtain w.h.p.

$$\|W_1\| \leq 6 \frac{1}{K} \sqrt{\frac{1}{p}} \sqrt{n} \leq \frac{1}{4},$$

where the last inequality follows from  $p \geq c \frac{n}{K^2}$  (cf. Lemma 2). In a similar manner, we obtain that w.h.p.

$$\|W_2\| \leq 6 \cdot 2 \sqrt{\frac{1-t}{p}} \lambda_{\mathcal{C}_{\mathcal{A}^c}} \cdot \sqrt{n} = 12 \sqrt{\frac{(1-t)}{p}} \cdot \frac{1}{48} \sqrt{\frac{t}{(1-t)n}} \cdot \sqrt{n} \leq \frac{1}{4},$$

where the last inequality follows from  $p \geq t$ , and w.h.p.

$$\|W_3\| \leq 6 \cdot 2 \sqrt{\frac{t}{1-q}} \lambda_{\mathcal{C}_{\mathcal{A}}} \cdot \sqrt{n} = 12 \sqrt{\frac{t}{1-q}} \cdot \frac{1}{48} \sqrt{\frac{1-t}{tn}} \cdot \sqrt{n} \leq \frac{1}{4},$$

where the last inequality follows from  $1-t \leq 1-q$ . Finally, since  $W_4 = (1+\epsilon)\lambda_{\mathcal{C}_{\mathcal{A}}} P_{R^c}(I)$  is a diagonal matrix, we have

$$\|W_4\| \leq (1+\epsilon)\lambda_{\mathcal{C}_{\mathcal{A}}} \leq 2 \cdot \frac{1}{48} \sqrt{\frac{1-t}{tn}} \leq \frac{1}{4}$$

since  $t \geq c \frac{1}{n}$  (cf. Lemma 2). We conclude that  $\|W\| \leq \sum_{i=1}^4 \|W_i\| \leq 1$ .

**(2) Bounding  $\|P_T W\|_\infty$ .**

Define the sets  $R_m := \{(i, j) : i, j \text{ in cluster } m\}$ , and recall that  $r$  is the number of clusters and  $R := \text{support}(Y^*) = \cup_{m=1}^r R_m$ . We have  $Y^* = \sum_{m=1}^r P_{R_m}(\mathbf{1}\mathbf{1}^\top)$ , and thus its singular vectors satisfies

$$U_0 U_0^\top = \sum_{m=1}^r \frac{1}{k_m} P_{R_m}(\mathbf{1}\mathbf{1}^\top).$$

Therefore, for  $i = 1, 2, 3$ , each entry of the matrix  $U_0 U_0^\top W_i$  equals  $\frac{1}{k_m}$  times the sum of  $k_m$  independent zero-mean random variables (which are the entries of  $W_i$ ), whose absolute values and variance are bounded in Lemma 3. We may use the standard Bernstein inequality (Lemma 5 in the Appendix) to bound each  $\|U_0 U_0^\top W_i\|_\infty$ .

For  $W_1$ , we have

$$\|U_0 U_0^\top W_1\|_\infty \leq \frac{1}{K} \cdot c_3 \sqrt{\frac{1}{pK^2}} \sqrt{K \log n} = c_3 \frac{1}{K} \sqrt{\frac{\log n}{pK}} \leq \frac{\log^2 n}{24^2} \sqrt{\frac{1}{Kn}},$$

where we use  $p \geq c \frac{n}{K^2}$  in the last inequality (c.f. Lemma 2). Similarly,  $W_2$  satisfies

$$\begin{aligned} \|U_0 U_0^\top W_2\|_\infty &\leq \frac{1}{K} \cdot c_3 \sqrt{\frac{1-t}{p}} \lambda_{c_{\mathcal{A}^c}} \sqrt{K \log n} \\ &= c_3 \sqrt{\frac{(1-t) \log n}{pK}} \frac{1}{48} \sqrt{\frac{t}{(1-t)n}} \leq \frac{\log^2 n}{24^2} \sqrt{\frac{1}{Kn}}, \end{aligned}$$

where we use  $p \geq t$  and  $\log n$  being sufficiently large in the last inequality. The matrix  $W_3$  obeys

$$\begin{aligned} \|U_0 U_0^\top W_3\|_\infty &\leq \frac{1}{K} \cdot c_3 \sqrt{\frac{t}{1-q}} \lambda_{c_{\mathcal{A}}} \sqrt{K \log n} \\ &= c_3 \sqrt{\frac{t \log n}{(1-q)K}} \frac{1}{48} \sqrt{\frac{1-t}{tn}} \leq \frac{\log^2 n}{24^2} \sqrt{\frac{1}{Kn}}, \end{aligned}$$

where we use  $1-q \geq 1-t$  and  $\log n$  being sufficiently in the last inequality. Finally, since  $W_4$  is a diagonal matrix supported on  $R^c$  and  $U_0 U_0^\top$  is supported on  $R$ , we have  $U_0 U_0^\top W_4 = 0$ .

On the other hand, we have

$$\lambda_{c_{\mathcal{A}}} \epsilon \geq \frac{1}{48} \sqrt{\frac{1-t}{tn}} \cdot 48 \sqrt{\frac{\log^4 n}{Kt(1-t)}} = \frac{1}{t} \sqrt{\frac{\log^4 n}{Kn}} \geq \frac{1}{24} \sqrt{\frac{\log^4 n}{Kn}}$$

and

$$\lambda_{c_{\mathcal{A}^c}} \epsilon \geq \frac{1}{48} \sqrt{\frac{t}{(1-t)n}} \cdot 48 \sqrt{\frac{\log^4 n}{Kt(1-t)}} = \frac{1}{(1-t)} \sqrt{\frac{\log^4 n}{Kn}} \geq \frac{1}{24} \sqrt{\frac{\log^4 n}{Kn}},$$

so  $\frac{1}{24} \epsilon \lambda \min\{c_{\mathcal{A}}, c_{\mathcal{A}^c}\} \geq \frac{\log^2 n}{24^2} \sqrt{\frac{1}{Kn}}$ . Combining with the previous bounds on  $\|U_0 U_0^\top W_i\|_\infty$ , we obtain  $\|(U_0 U_0^\top W_i)\|_\infty \leq \frac{1}{24} \epsilon \lambda \min\{c_{\mathcal{A}}, c_{\mathcal{A}^c}\}$ .

Now observe that since  $W$  and  $U_0 U_0^\top$  are both symmetric, we have  $W U_0 U_0^\top = (U_0 U_0^\top W)^\top$  and thus  $\|W U_0 U_0^\top\|_\infty = \|U_0 U_0^\top W\|_\infty$ . Furthermore, we have

$$U_0 U_0^\top W U_0 U_0^\top = U_0 U_0^\top \sum_{m=1}^r \frac{1}{k_m} P_{R_m} (\mathbf{1}\mathbf{1}^\top),$$

which implies  $\|U_0 U_0^\top W U_0 U_0^\top\|_\infty \leq \|U_0 U_0^\top W\|_\infty$ . It follows that

$$\begin{aligned} \|P_T W\|_\infty &= \|U_0 U_0^\top W + W U_0 U_0^\top - U_0 U_0^\top W U_0 U_0^\top\|_\infty \\ &\leq \|U_0 U_0^\top W\|_\infty + \|W U_0 U_0^\top\|_\infty + \|U_0 U_0^\top W U_0 U_0^\top\|_\infty \\ &\leq 3 \|U_0 U_0^\top W\|_\infty \leq 3 \sum_{i=1}^4 \|U_0 U_0^\top W_i\|_\infty. \end{aligned}$$

Using the bounds on  $\|U_0 U_0^\top W_i\|_\infty$  derived above, we obtain that  $\|P_T W\|_\infty \leq 12 \cdot \frac{1}{24} \epsilon \lambda \min\{c_{\mathcal{A}}, c_{\mathcal{A}^c}\} = \frac{1}{2} \epsilon \lambda \min\{c_{\mathcal{A}}, c_{\mathcal{A}^c}\}$ .

(3) The two equalities in condition (c) in Proposition 1 hold by the definition of  $W$ . We now turn to the inequalities in condition (c). Because  $1 - q \geq 1 - t$  and  $p \leq 4t$ , we have  $\frac{1-q}{p} \geq \frac{1}{4} \frac{1-t}{t}$ . It follows from the conditions in Theorem 1 that

$$\begin{aligned} \frac{p-q}{4} &\geq c \sqrt{p(1-q)} \max \left\{ \sqrt{n}/K, \sqrt{\log^4 n/K} \right\} \\ &\geq 8p(1-t) \cdot \frac{48}{\sqrt{t(1-t)}} \max \left\{ \sqrt{n}/K, \sqrt{\log^4 n/K} \right\} = 8p(1-t)\epsilon. \end{aligned} \quad (12)$$

We thus have

$$p - t \geq p - \left( \frac{3}{4}p + \frac{1}{4}q \right) = \frac{p-q}{4} \geq 8p(1-t)\epsilon.$$

One verifies that this implies  $(1+\epsilon)\sqrt{\frac{t}{1-t}} \frac{1-p}{p} \leq (1-2\epsilon)\sqrt{\frac{1-t}{t}}$ . Plugging in the values of  $c_{\mathcal{A}}$  and  $c_{\mathcal{A}^c}$  in (4) yields

$$(1+\epsilon) \frac{c_{\mathcal{A}^c}(1-p)}{p} \leq (1-2\epsilon)c_{\mathcal{A}},$$

Hence, for each  $(i, j) \in R \cap \mathcal{A}$ , we have

$$(U_0 U_0^\top + W)_{ij} = \frac{1}{p}(U_0 U_0^\top)_{ij} + (1+\epsilon)\lambda c_{\mathcal{A}^c} \frac{1-p}{p} \leq \frac{1}{p}(U_0 U_0^\top)_{ij} + (1-2\epsilon)c_{\mathcal{A}}.$$

We also have

$$\frac{1}{p}(U_0 U_0^\top)_{ij} \leq \frac{1}{pK} \stackrel{(i)}{\leq} \frac{48}{K} \sqrt{\frac{n}{t(1-t)}} \cdot \frac{1}{48} \sqrt{\frac{1-t}{tn}} \leq \epsilon \cdot \lambda c_{\mathcal{A}},$$

where (i) follows from  $p \geq t$ . Combining the last two displays proves the first inequality in condition (c).

Similarly, we have

$$t - q \geq \left( \frac{p}{4} + \frac{3q}{4} \right) - q = \frac{p-q}{4} \stackrel{(ii)}{\geq} 8p(1-t)\epsilon \stackrel{(iii)}{\geq} 2t(1-q)\epsilon,$$

where (ii) follows from (12) and (iii) follows from  $p \geq t$  and  $1-t \geq 1 - \frac{3}{4}p - \frac{1}{4}q \geq \frac{1}{4}(1-q)$ . This implies  $(1+\epsilon)\sqrt{\frac{1-t}{t}} \frac{q}{1-q} \leq (1-\epsilon)\sqrt{\frac{t}{1-t}}$ . Therefore, for each  $(i, j) \in R^c \cap \mathcal{A}^c$ ,

$$w_{ij} = -(1+\epsilon) \frac{c_{\mathcal{A}} q}{1-q} \geq -(1-\epsilon)c_{\mathcal{A}^c},$$

proving the second inequality in condition (c). This completes the proof of Proposition 2.

## VIII. PROOF OF THEOREM 2

We use a standard information theoretic argument via Fano's inequality. For simplicity we assume  $n_1/K$  and  $n_2/K$  are both integers, and we use  $c_1, c_2, \dots$  to denote positive absolute constants. Let  $\mathcal{F}$  be the set of all possible ways of assigning  $n$  nodes into  $n_1/K$  clusters of equal size  $K$ . When  $K = \Theta(n_1) = \Theta(n_2)$ , the cardinality of  $\mathcal{F}$  can be bounded as

$$M := |\mathcal{F}| = \frac{1}{(n_1/K)!} \binom{n}{K} \binom{n-K}{K} \cdots \binom{n_1+K}{K} \geq c_2 \cdot c_1^{\frac{1}{2}n}$$

for some  $c_1 > 1$  and  $c_2 > 0$ .

Suppose the true cluster matrix  $Y^*$  is obtained uniformly at random from  $\mathcal{F}$ , and the graph  $A$  is generated from  $Y^*$  according to GSBM with uniform edge probabilities. We use  $\mathbb{P}_{A|Y^*}$  to denote the distribution of  $A$  given  $Y^*$ . Let  $\hat{Y}$  be any measurable function of  $A$ . The standard Fano's inequality gives

$$\sup_{Y^* \in \mathcal{F}} \mathbb{P} [\hat{Y} \neq Y^* | Y^*] \geq 1 - \frac{I(A; Y^*) + \log 2}{\log M} \geq 1 - \frac{I(A; Y^*) + \log 2}{c_3 n}$$

for  $n$  is sufficiently large. We now bound the mutual information  $I(A; Y^*)$ . Observe that

$$\begin{aligned} I(A; Y^*) &= H(A) - H(A|Y^*) \leq \sum_{(i,j): i > j} H(a_{ij}) - H(A|Y^*) \\ &= \binom{n}{2} H(a_{12}) - \binom{n}{2} H(a_{12}|Y^*) = \binom{n}{2} I(a_{12}; Y^*), \end{aligned}$$

where in the second equality we have used the symmetry under the uniform distribution of  $Y^*$  and the conditional independence between  $a'_{ij}$ s. By definition of the mutual information, we have

$$I(a_{12}; Y^*) = I(a_{12}; y_{12}^*) = \mathbb{E}_{y_{12}^*} [D(\mathbb{P}(a_{12}|y_{12}^*) || \mathbb{P}(a_{12}))].$$

We compute the divergence on the last RHS. Let  $\alpha := \mathbb{P}(y_{12}^* = 1) = \frac{(K-1)n_1}{n^2}$  and  $\gamma := \mathbb{P}(a_{11} = 1) = \alpha p + (1 - \alpha)q$ . It follows that

$$\begin{aligned} &\mathbb{E}_{y_{12}^*} [D(\mathbb{P}(a_{12}|y_{12}^*) || \mathbb{P}(a_{12}))] \\ &= \sum_{y \in \{0,1\}} \sum_{a \in \{0,1\}} \mathbb{P}(y_{12}^* = y) \mathbb{P}(a_{12} = a | y_{12}^* = y) \log \frac{\mathbb{P}(a_{12} = a | y_{12}^* = y)}{\mathbb{P}(a_{12} = a)} \\ &= \alpha p \log \frac{p}{\gamma} + \alpha(1-p) \log \frac{(1-p)}{(1-\gamma)} + (1-\alpha)q \log \frac{q}{\gamma} + (1-\alpha)(1-q) \log \frac{(1-q)}{(1-\gamma)} \\ &\leq \alpha p \left( \frac{p}{\gamma} - 1 \right) + \alpha(1-p) \left( \frac{1-p}{1-\gamma} - 1 \right) + (1-\alpha)q \left( \frac{q}{\gamma} - 1 \right) + (1-\alpha)(1-q) \left( \frac{1-q}{1-\gamma} - 1 \right) \\ &= \frac{\alpha(1-\alpha)(p-q)^2}{\gamma(1-\gamma)} \leq c_4 \frac{(p-q)^2}{p(1-q)}, \end{aligned}$$

where in the last inequality we use  $\gamma \geq \alpha p$ ,  $1 - \gamma \geq (1 - \alpha)(1 - q)$  and  $\alpha, 1 - \alpha = \Theta(1)$ . Combining pieces, we obtain

$$\sup_{Y \in \mathcal{F}} \mathbb{P} [\hat{Y} \neq Y | Y] \geq 1 - \frac{c_5 \frac{(p-q)^2 n^2}{p(1-q)} + \log 2}{c_3 n}.$$

For the last R.H.S. to be less than  $\frac{1}{4}$ , we need  $\frac{(p-q)^2}{p(1-q)} \geq c_6 \frac{1}{n}$ . This completes the proof of the theorem.

## IX. PROOF OF THEOREM 3

Let  $\lambda_i$  be the  $i$ -th eigenvalue of the matrix  $\mathbb{E}[A]$  (counting multiplicity). Observe that the matrix  $\bar{A} := A - \mathbb{E}A$  is a random symmetric matrix with independent zero-mean entries, each of which is bounded in absolute value by 1 and has variance bounded by  $p(1-p) \vee q(1-q) \leq p(1-q)$ . Under the condition of Theorem 3, we may apply Lemma 4 to obtain  $\|\bar{A}\| \leq 4\sqrt{p(1-q)n}$  w.h.p. It then follows from Weyl's inequality [40] that w.h.p.

$$\max_i \left\{ \left| \hat{\lambda}_i - \lambda_i \right| \right\} \leq \|A - \mathbb{E}A\| = \|\bar{A}\| \leq 4\sqrt{p(1-q)n}. \quad (13)$$

In the sequel, we assume we are on the event that (13) holds.

a) *Estimation of  $r$* : Recall that  $\lambda_1 = K(p - q) + nq + (1 - p)$ ,  $\lambda_2, \dots, \lambda_r = K(p - q) + (1 - p)$ , and  $\lambda_{r+1}, \dots, \lambda_n = 1 - p$ . The inequality (13) implies that for some universal constant  $c_1$ ,

- $\hat{\lambda}_1 - \hat{\lambda}_2 \leq \lambda_1 - \lambda_2 + |\hat{\lambda}_1 - \lambda_1| + |\hat{\lambda}_2 - \lambda_2| \leq nq + c_1 \sqrt{p(1 - q)n}$ ;
- $\hat{\lambda}_i - \hat{\lambda}_{i+1} \leq \lambda_i - \lambda_{i+1} + |\hat{\lambda}_i - \lambda_i| + |\hat{\lambda}_{i+1} - \lambda_{i+1}| \leq c_1 \sqrt{p(1 - q)n}$  for  $i = 2, \dots, r - 1$  and  $i \geq r + 1$ ;
- $\hat{\lambda}_r - \hat{\lambda}_{r+1} \geq \lambda_r - \lambda_{r+1} - |\hat{\lambda}_r - \lambda_r| - |\hat{\lambda}_{r+1} - \lambda_{r+1}| \geq K(p - q) - c_1 \sqrt{p(1 - q)n}$ .

Under the condition of Theorem 3, we have  $K(p - q) \geq c_2 \sqrt{p(1 - q)n}$  for some constant  $c_2$ . This implies  $\hat{\lambda}_r - \hat{\lambda}_{r+1} > \frac{K(p - q)}{2} > \hat{\lambda}_i - \hat{\lambda}_{i+1}$  for all  $i > 1$  and  $i \neq r$ . This guarantees  $\hat{r} = r$  and thus  $\hat{K} = K$ .

*Estimation of  $p$  and  $q$* : By (13), the estimation error of  $\hat{q}$  satisfies

$$|\hat{q} - q| = \left| \frac{\hat{\lambda}_1 - \lambda_1}{n} - \frac{\hat{\lambda}_2 - \lambda_2}{n} \right| \leq \frac{|\hat{\lambda}_1 - \lambda_1| + |\hat{\lambda}_2 - \lambda_2|}{n} \leq c_3 \frac{\sqrt{p(1 - q)n}}{K}.$$

Similarly, we have

$$\begin{aligned} |\hat{p} - p| &= \left| \frac{\hat{K}\hat{\lambda}_1 + (n - \hat{K})\hat{\lambda}_2 - n}{n(\hat{K} - 1)} - \frac{K\lambda_1 + (n - K)\lambda_2 - n}{n(K - 1)} \right| \\ &= \left| \frac{K(\hat{\lambda}_1 - \lambda_1) + (n - K)(\hat{\lambda}_2 - \lambda_2)}{n(K - 1)} \right| \leq c_3 \frac{\sqrt{p(1 - q)n}}{K}. \end{aligned}$$

b) *Choosing  $t$* : Using the above bounds on  $\hat{p}$  and  $\hat{q}$ , we obtain

$$\begin{aligned} t &= \frac{p + q}{2} + \frac{\hat{p} - p + \hat{q} - q}{2} \leq \frac{p + q}{2} + c_4 \frac{\sqrt{p(1 - q)n}}{K} \\ &\leq \frac{p + q}{2} + \frac{p - q}{4} = \frac{3}{4}p + \frac{1}{4}q, \end{aligned}$$

where in the last inequality we use the assumption  $\frac{p - q}{4} \geq c_4 \frac{\sqrt{p(1 - q)n}}{K}$  in the theorem. This proves one side of inequality for  $t$ , and the other side is proved in a similar way.

## X. CONCLUSION

This work is motivated by clustering large-scale networks such as modern online social networks, where the graphs are often highly noisy and has heterogeneous and non-random structures. We considered a natural and versatile model, namely the semi-random Generalized Stochastic Blockmodel, for clustered random graphs. This model recovers many classical generative models for graph clustering. We presented a convex optimization formulation, essentially a convexification of the maximum likelihood estimator. Our theoretic analysis shows that this method is guaranteed to recover the correct clusters under a wide range of problem parameters of the problem. In fact, our method outperforms, i.e., succeeds under less restrictive conditions, every existing method in this setting. Simulation studies also validates the effectiveness of the proposed method. Immediate goals for future work include faster algorithm implementations, as well as developing effective post-processing schemes (e.g., rounding) when the obtained solution is not an exact cluster matrix.

## REFERENCES

- [1] S. Fortunato, “Community detection in graphs,” *Phys. Rep.*, vol. 486, no. 3-5, pp. 75–174, 2010.
- [2] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney, “Statistical properties of community structure in large social and information networks,” in *Proceeding of the 17th international conference on World Wide Web*. ACM, 2008, pp. 695–704.



- [3] A. Condon and R. Karp, “Algorithms for graph partitioning on the planted partition model,” *Random Structures Algorithms*, vol. 18, no. 2, pp. 116–140, 2001.
- [4] P. Holland, K. Laskey, and S. Leinhardt, “Stochastic blockmodels: Some first steps,” *Social Networks*, vol. 5, pp. 109–137, 1983.
- [5] F. McSherry, “Spectral partitioning of random graphs,” in *Proceedings of 42nd IEEE Symposium on Foundations of Computer Science*. IEEE, 2001, pp. 529–537.
- [6] B. Ames and S. Vavasis, “Convex optimization for the planted k-disjoint-clique problem,” *Arxiv preprint arXiv:1008.2814*, 2011.
- [7] N. Alon and N. Kahale, “A spectral technique for coloring random 3-colorable graphs,” *SIAM J. Comput.*, vol. 26, no. 6, pp. 1733–1748, 1997.
- [8] A. Coja-Oghlan, “Coloring semirandom graphs optimally,” *Automata, Languages and Programming*, pp. 71–100, 2004.
- [9] B. Bollobás and A. Scott, “Max cut for random graphs with a planted partition,” *Combin. Probab. Comput.*, vol. 13, no. 4-5, pp. 451–474, 2004.
- [10] U. Feige and J. Kilian, “Heuristics for semirandom graph problems,” *J. Comput. System Sci.*, vol. 63, no. 4, pp. 639–671, 2001.
- [11] M. Krivelevich and D. Vilenchik, “Semirandom models as benchmarks for coloring algorithms,” in *Third Workshop on Analytic Algorithmics and Combinatorics (ANALCO)*, 2006, pp. 211–221.
- [12] K. Chaudhuri, F. Chung, and A. Tsiatas, “Spectral clustering of graphs with general degrees in the extended planted partition model,” *J. Mach. Learn. Res.*, 2012.
- [13] A. Decelle, F. Krzakala, C. Moore, and L. Zdeborová, “Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications,” *Phys. Rev. E*, vol. 84, no. 6, p. 066106, 2011.
- [14] R. R. Nadakuditi and M. Newman, “Graph spectra and the detectability of community structure in networks,” *Phys. Rev. Lett.*, vol. 108, no. 18, p. 188701, 2012.
- [15] R. Boppana, “Eigenvalues and graph bisection: An average-case analysis,” in *28th Annual Symposium on Foundations of Computer Science, 1987*. IEEE, 1987, pp. 280–285.
- [16] M. Jerrum and G. Sorkin, “The metropolis algorithm for graph bisection,” *Discrete Appl. Math.*, vol. 82, no. 1-3, pp. 155–175, 1998.
- [17] T. Carson and R. Impagliazzo, “Hill-climbing finds random planted bisections,” in *Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete algorithms*. SIAM, 2001, pp. 903–909.
- [18] J. Giesen and D. Mitsche, “Reconstructing many partitions using spectral techniques,” in *Fundamentals of Computation Theory*. Springer, 2005, pp. 433–444.
- [19] R. Shamir and D. Tsur, “Improved algorithms for the random cluster graph model,” *Random Structures Algorithms*, vol. 31, no. 4, pp. 418–449, 2007.
- [20] A. Coja-Oghlan, “Graph partitioning via adaptive spectral techniques,” *Combin. Probab. Comput.*, vol. 19, no. 2, p. 227, 2010.
- [21] K. Rohe, S. Chatterjee, and B. Yu, “Spectral clustering and the high-dimensional stochastic blockmodel,” *Ann. Statist.*, vol. 39, no. 4, pp. 1878–1915, 2011.
- [22] S. Oymak and B. Hassibi, “Finding dense clusters via “Low Rank+ Sparse” decomposition,” *Arxiv preprint arXiv:1104.5186*, 2011.
- [23] B. Ames, “Guaranteed clustering and biclustering via semidefinite programming,” *Arxiv preprint arXiv:1202.3663*, 2012.
- [24] V. Chandrasekaran, S. Sanghavi, S. Parrilo, and A. Willsky, “Rank-sparsity incoherence for matrix decomposition,” *SIAM J. Optim.*, vol. 21, no. 2, pp. 572–596, 2011.
- [25] E. J. Candès, X. Li, Y. Ma, and J. Wright, “Robust principal component analysis?” *J. ACM*, vol. 58, no. 3, p. 11, 2011.
- [26] Y. Chen, A. Jalali, S. Sanghavi, and C. Caramanis, “Low-rank matrix recovery from errors and erasures,” *IEEE Trans. Inform. Theory*, vol. 59, no. 7, pp. 4324–4337, 2013.
- [27] X. Li, “Compressed sensing and matrix completion with constant proportion of corruptions,” *Constr.*

- Approx.*, vol. 37, no. 1, pp. 73–99, 2013.
- [28] Y. Chen, A. Jalali, S. Sanghavi, and H. Xu, “Clustering partially observed graphs via convex optimization,” *Arxiv preprint arXiv:1104.4803*, 2011.
  - [29] B. Recht, M. Fazel, and P. Parrilo, “Guaranteed Minimum-Rank Solutions of Linear Matrix Equations via Nuclear Norm Minimization,” *SIAM Rev.*, vol. 52, no. 471, 2010.
  - [30] C. Mathieu and W. Schudy, “Correlation clustering with noisy input,” in *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 2010, pp. 712–728.
  - [31] A. Frieze and C. McDiarmid, “Algorithmic theory of random graphs,” *Random Structures Algorithms*, vol. 10, no. 1-2, pp. 5–42, 1997.
  - [32] N. Alon, M. Krivelevich, and B. Sudakov, “Finding a large hidden clique in a random graph,” *Random Structures Algorithms*, vol. 13, no. 3-4, pp. 457–466, 1998.
  - [33] E. Hazan and R. Krauthgamer, “How hard is it to approximate the best nash equilibrium?” *SIAM J. Comput.*, vol. 40, no. 1, pp. 79–91, 2011.
  - [34] A. Juels and M. Peinado, “Hiding cliques for cryptographic security,” *Des. Codes Cryptogr.*, vol. 20, no. 3, pp. 269–280, 2000.
  - [35] A. Anandkumar, R. Ge, D. Hsu, and S. Kakade, “A tensor approach to learning mixed membership community models,” in *The 26th Conference on Learning Theory*, 2013.
  - [36] Y. Chen, S. Sanghavi, and H. Xu, “Clustering sparse graphs,” in *Advances in Neural Information Processing Systems 25 (NIPS)*, 2012.
  - [37] Z. Lin, M. Chen, L. Wu, and Y. Ma, “The Augmented Lagrange Multiplier Method for Exact Recovery of Corrupted Low-Rank Matrices,” *UIUC Technical Report UILU-ENG-09-2215*, 2009.
  - [38] R. Sibson, “Slink: an optimally efficient algorithm for the single-link cluster method,” *The Computer Journal*, vol. 16, no. 1, pp. 30–34, 1973.
  - [39] U. Von Luxburg, “A tutorial on spectral clustering,” *Stat. Comput.*, vol. 17, no. 4, pp. 395–416, 2007.
  - [40] R. Bhatia, *Perturbation bounds for matrix eigenvalues*. Longman, Harlow, 1987.
  - [41] D. Achlioptas and F. Mcsherry, “Fast computation of low-rank matrix approximations,” *J. ACM*, vol. 54, no. 2, p. 9, 2007.
  - [42] R. Vershynin, “Introduction to the non-asymptotic analysis of random matrices,” *Arxiv preprint arxiv:1011.3027*, 2010.

## APPENDIX

In this section, we record two technical lemmas that are needed in the proofs of our theoretical results. The first lemma is a standard bound on the spectral norm of a random symmetric matrix.

**Lemma 4.** *Suppose  $Y$  is a symmetric  $n \times n$  matrix, where  $Y_{ij}$ ,  $1 \leq i, j \leq n$  are independent random variables, each of which has mean 0 and variance at most  $\sigma^2$  and is bounded in absolute value by  $B$ . If  $\sigma \geq c_1 \frac{B \log^2 n}{\sqrt{n}}$  for some absolute constant  $c_1 > 0$ , then with probability at least  $1 - n^{-10}$ ,*

$$\|Y\| \leq 4\sigma\sqrt{n}.$$

*Proof.* Except for  $Y$  being symmetric, the proof is the same that of Theorem 3.1 in [41]. □

The second lemma is the standard Bernstein inequality for the sum of independent random variables.

**Lemma 5.** ([42], Proposition 5.16) *Let  $Y_1, \dots, Y_N$  be independent random variables, each of which is bounded in absolute value by  $B$  a.s. and has variance bounded by  $\sigma^2$ . There exist universal positive constants  $c_0, c_1, c_2$  independent of  $\sigma, B, N$  and  $n$  such that if  $\sigma \geq B\sqrt{\frac{\log n}{N}}$ , then we have*

$$\left| \sum_{i=1}^N Y_i - \mathbb{E} \left[ \sum_{i=1}^N Y_i \right] \right| \leq c_0 \sigma \sqrt{N \log n}$$

*with probability at least  $1 - c_1 n^{-c_2}$ .*